Visualization Analysis & Design

Tamara Munzner
Department of Computer Science
University of British Columbia

Data Visualization Masterclass: Principles, Tools, and Storytelling
June 13 2017, VIZBI/VIVID, Sydney Australia

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
@tamaramunzner
Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
Defining visualization (vis)

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Why?...
Why have a human in the loop?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

Visualization is suitable when there is a need to augment human capabilities rather than replace people with computational decision-making methods.

• don’t need vis when fully automatic solution exists and is trusted
• many analysis problems ill-specified
  – don’t know exactly what questions to ask in advance
• possibilities
  – long-term use for end users (e.g. exploratory analysis of scientific data)
  – presentation of known results
  – stepping stone to better understanding of requirements before developing models
  – help developers of automatic solution refine/debug, determine parameters
  – help end users of automatic solutions verify, build trust
Why use an external representation?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

• external representation: replace cognition with perception

Why represent all the data?

Computer-based visualization systems provide visual representations of datasets designed to help people carry out tasks more effectively.

- summaries lose information, details matter
  - confirm expected and find unexpected patterns
  - assess validity of statistical model

Anscombe’s Quartet

<table>
<thead>
<tr>
<th>Identical statistics</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>x mean</td>
<td>9</td>
</tr>
<tr>
<td>x variance</td>
<td>10</td>
</tr>
<tr>
<td>y mean</td>
<td>7.5</td>
</tr>
<tr>
<td>y variance</td>
<td>3.75</td>
</tr>
<tr>
<td>x/y correlation</td>
<td>0.816</td>
</tr>
</tbody>
</table>

https://www.youtube.com/watch?v=DbJyPELmhJc

Same Stats, Different Graphs
Why are there resource limitations?

Vis designers must take into account three very different kinds of resource limitations: those of computers, of humans, and of displays.

• computational limits
  – processing time
  – system memory

• human limits
  – human attention and memory

• display limits
  – pixels are precious resource, the most constrained resource
  – information density: ratio of space used to encode info vs unused whitespace
    • tradeoff between clutter and wasting space, find sweet spot between dense and sparse
Analysis framework: Four levels, three questions

- **domain** situation
  - who are the target users?

- **abstraction**
  - translate from specifics of domain to vocabulary of vis

- **what** is shown? data abstraction

- **why** is the user looking at it? task abstraction

- **idiom**

- **how** is it shown?
  - visual encoding idiom: how to draw
  - interaction idiom: how to manipulate

- **algorithm**
  - efficient computation

---


Validation methods from different fields for each level

- **Domain situation**
  - Observe target users using existing tools

- **Data/task abstraction**
  - **Visual encoding/interaction idiom**
    - Justify design with respect to alternatives
  - **Algorithm**
    - Measure system time/memory
    - Analyze computational complexity
  - Analyze results qualitatively
  - Measure human time with lab experiment *(lab study)*
  - Observe target users after deployment *(field study)*
  - Measure adoption

- **anthropology/ethnography**

- **design**

- **computer science**

- **cognitive psychology**

- mismatch: cannot show idiom good with system timings
- mismatch: cannot show abstraction good with lab study
Why analyze?

- imposes a structure on huge design space
  - scaffold to help you think systematically about choices
  - analyzing existing as stepping stone to designing new

---

**SpaceTree**

- Encode
- Navigate
- Select
- Filter
- Aggregate

**TreeJuxtaposer**

- Encode
- Navigate
- Select
- Arrange

---


Datasets

- **Data Types**
  - Items
  - Attributes
  - Links
  - Positions
  - Grids

- **Data and Dataset Types**
  - **Tables**
    - Items
    - Attributes
  - **Networks & Trees**
    - Items (nodes)
    - Grids
    - Positions
    - Attributes
  - **Geometry**
    - Items
    - Positions
    - Attributes
  - **Clusters, Sets, Lists**
    - Items

- **Dataset Types**
  - **Tables**
  - **Networks**
  - **Fields (Continuous)**
  - **Geometry (Spatial)**

- **Dataset Availability**
  - **Static**
  - **Dynamic**

- **Attribute Types**
  - **Categorical**
    - +
    - ●
    - □
    - ▲
  - **Ordered**
    - Ordinal
  - **Quantitative**
  - **Ordering Direction**
    - Sequential
    - Diverging
    - Cyclic
Dataset and data types

**Dataset Types**
- **Tables**
- **Networks**

**Attribute Types**
- **Categorical**
- **Ordered**
  - **Ordinal**
  - **Quantitative**

**Spatial**
- **Fields (Continuous)**
- **Geometry (Spatial)**
• \{action, target\} pairs
  – discover distribution
  – compare trends
  – locate outliers
  – browse topology
Actions 1: Analyze

- **consume**
  - discover vs present
    - classic split
    - aka explore vs explain
  - enjoy
    - newcomer
    - aka casual, social

- **produce**
  - annotate, record
  - derive
    - crucial design choice
Actions II: Search

• what does user know? ➔ Search

– target, location

<table>
<thead>
<tr>
<th></th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location known</td>
<td>Lookup</td>
<td>Browse</td>
</tr>
<tr>
<td>Location unknown</td>
<td>Locate</td>
<td>Explore</td>
</tr>
</tbody>
</table>
Actions III: Query

• what does user know? ➔ Search
  – target, location

• how much of the data matters?
  – one, some, all

<table>
<thead>
<tr>
<th>Location known</th>
<th>Target known</th>
<th>Target unknown</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lookup</td>
<td></td>
<td>Browse</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location unknown</th>
<th>Identify</th>
<th>Compare</th>
<th>Summarize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Locate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Explore</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Targets

All Data
- Trends
- Outliers
- Features

Attributes
- One
  - Distribution
  - Extremes
- Many
  - Dependency
  - Correlation
  - Similarity

Network Data
- Topology
  - Paths

Spatial Data
- Shape
How?

**Encode**

- **Arrange**
  - Express
  - Separate
- **Order**
  - Align
- **Use**

**Map**

from **categorical** and **ordered** attributes

- **Color**
  - Hue
  - Saturation
  - Luminance
- **Size, Angle, Curvature, ...**

- **Shape**
  - + ● ■ △

- **Motion**
  - Direction, Rate, Frequency, ...

**Manipulate**

- **Change**
- **Select**
- **Navigate**

**Facet**

- **Juxtapose**
- **Partition**
- **Superimpose**

**Reduce**

- **Filter**
- **Aggregate**
- **Embed**

**What?**

**Why?**

**How?**
Further reading

  – Chap 1: What’s Vis, and Why Do It?
  – Chap 2: What: Data Abstraction
  – Chap 3: Why: Task Abstraction


Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
Encoding visually

• analyze idiom structure
Definitions: Marks and channels

• marks
  – geometric primitives

• channels
  – control appearance of marks
Encoding visually with marks and channels

• analyze idiom structure
  – as combination of marks and channels

1: vertical position
mark: line

2: vertical position
horizontal position
mark: point

3: vertical position
horizontal position
color hue
mark: point

4: vertical position
horizontal position
color hue
size (area)
mark: point
Channels

- **Position on common scale**
- **Position on unaligned scale**
- **Length (1D size)**
- **Tilt/angle**
- **Area (2D size)**
- **Depth (3D position)**
- **Color luminance**
- **Color saturation**
- **Curvature**
- **Volume (3D size)**

**Spatial region**

**Color hue**

**Motion**

**Shape**
Channels: Rankings

**Magnitude Channels: Ordered Attributes**
- Position on common scale
- Position on unaligned scale
- Length (1D size)
- Tilt/angle
- Area (2D size)
- Depth (3D position)
- Color luminance
- Color saturation
- Curvature
- Volume (3D size)

**Identity Channels: Categorical Attributes**
- Spatial region
- Color hue
- Motion
- Shape

- **effectiveness principle**
  - encode most important attributes with highest ranked channels
- **expressiveness principle**
  - match channel and data characteristics
Accuracy: Fundamental Theory

Steven’s Psychophysical Power Law: $S = I^N$
Accuray: Vis experiments

Discriminability: How many usable steps?

• must be sufficient for number of attribute levels to show
  – linewidth: few bins

[mappa.mundi.net/maps/maps 014/telegeography.html]
Separability vs. Integrality

<table>
<thead>
<tr>
<th>Category</th>
<th>Position</th>
<th>Size</th>
<th>Width</th>
<th>Red</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Hue (Color)</td>
<td>+ Hue (Color)</td>
<td>+ Height</td>
<td>+ Green</td>
<td></td>
</tr>
<tr>
<td>Fully separable</td>
<td>Some interference</td>
<td>Some/significant interference</td>
<td>Major interference</td>
<td></td>
</tr>
</tbody>
</table>

- 2 groups each
- 2 groups each
- 3 groups total: integral area
- 4 groups total: integral hue
Further reading

  – Chap 5: Marks and Channels


Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmmtalks.html#vad17sydney
<table>
<thead>
<tr>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Encode</strong></td>
</tr>
<tr>
<td>➔ Arrange</td>
</tr>
<tr>
<td>➔ Express</td>
</tr>
<tr>
<td>➔ Separate</td>
</tr>
<tr>
<td>➔ Order</td>
</tr>
<tr>
<td>➔ Align</td>
</tr>
<tr>
<td>➔ Use</td>
</tr>
</tbody>
</table>

**Map**
- from *categorical* and *ordered* attributes
- ➔ Color
  - ➔ Hue
  - ➔ Saturation
  - ➔ Luminance
- ➔ Size, Angle, Curvature, ...

**Facet**
- ➔ Juxtapose
- ➔ Partition
- ➔ Superimpose

**Reduce**
- ➔ Filter
- ➔ Aggregate
- ➔ Embed

---

**What?**

**Why?**

**How?**
Challenges of Color

• what is wrong with this picture?

@WTFViz
“visualizations that make no sense”

Categorical vs ordered color

Decomposing color

• first rule of color: do not talk about color!
  – color is confusing if treated as monolithic

• decompose into three channels
  – ordered can show magnitude
    • luminance
    • saturation
  – categorical can show identity
    • hue

• channels have different properties
  – what they convey directly to perceptual system
  – how much they can convey: how many discriminable bins can we use?
Luminance

• need luminance for edge detection
  – fine-grained detail only visible through luminance contrast
  – legible text requires luminance contrast!

• intrinsic perceptual ordering

Categorical color: limited number of discriminable bins

- human perception built on relative comparisons
  - great if color contiguous
  - surprisingly bad for absolute comparisons
- noncontiguous small regions of color
  - fewer bins than you want
  - rule of thumb: 6-12 bins, including background and highlights

[Cinteny: flexible analysis and visualization of synteny and genome rearrangements in multiple organisms. Sinha and Meller. BMC Bioinformatics, 8:82, 2007.]
ColorBrewer

- http://www.colorbrewer2.org
- saturation and area example: size affects salience!
Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable


Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – large-scale structure: fewer hues


Ordered color: Rainbow is poor default

• problems
  – perceptually unordered
  – perceptually nonlinear

• benefits
  – fine-grained structure visible and nameable

• alternatives
  – large-scale structure: fewer hues
  – fine structure: multiple hues with monotonically increasing luminance [eg viridis R/python]
Viridis

- colorful, perceptually uniform, colorblind-safe, monotonically increasing luminance

https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html
Ordered color: Rainbow is poor default

- problems
  - perceptually unordered
  - perceptually nonlinear
- benefits
  - fine-grained structure visible and nameable
- alternatives
  - large-scale structure: fewer hues
  - fine structure: multiple hues with monotonically increasing luminance [eg viridis R/python]
  - segmented rainbows for binned or categorical


Colormaps

- Categorical
- Ordered
  - Sequential
  - Diverging
- Bivariate

- Categorical limits: noncontiguous
  - 6-12 bins hue/color
    - far fewer if colorblind
  - 3-4 bins luminance, saturation
  - size heavily affects salience
    - use high saturation for small regions, low saturation for large

Further reading

  • Chap 10: Map Color and Other Channels

• ColorBrewer, Brewer.
  • http://www.colorbrewer2.org

  • http://www.stonesc.com/Vis06


Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Coffee Break** 10:30-10:50am

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
How?

**Encode**

- **Arrange**
  - Express
  - Separate
- **Order**
  - Align
- **Use**

**Map**

- from *categorical* and *ordered* attributes
- **Color**
  - Hue
  - Saturation
  - Luminance
- **Size, Angle, Curvature, ...**
- **Shape**
  - + • ■ △
- **Motion**
  - Direction, Rate, Frequency, ...

**Manipulate**

- **Change**

**Facet**

- **Juxtapose**

**Reduce**

- **Filter**
- **Aggregate**
- **Embed**

---

48
How to handle complexity: 3 more strategies

**Manipulate**

- **Change**
  - ![Change Diagram]

- **Select**
  - ![Select Diagram]

- **Navigate**
  - ![Navigate Diagram]

**Facet**

- **Juxtapose**
  - ![Juxtapose Diagram]

**Reduce**

- **Filter**
  - ![Filter Diagram]

- **Aggregate**
  - ![Aggregate Diagram]

- **Embed**
  - ![Embed Diagram]

- **Derive**
  - ![Derive Diagram]

- **Change view over time**
- **Facet across multiple views**
- **Reduce items/attributes within single view**
- **Derive new data to show within view**
How to handle complexity: 3 more strategies

- **Manipulate**
  - **Change**
  - **Select**
  - **Navigate**

- **Facet**
  - **Juxtapose**
  - **Partition**
  - **Superimpose**

- **Reduce**
  - **Filter**
  - **Aggregate**
  - **Embed**

**Derive**

- change over time
  - most obvious & flexible of the 4 strategies
Change over time

• change any of the other choices
  – encoding itself
  – parameters
  – arrange: rearrange, reorder
  – aggregation level, what is filtered...

• why change?
  – one of four major strategies
    • change over time
    • facet data by partitioning into multiple views
    • reduce amount of data shown within view
      – embedding focus + context together
  – most obvious, powerful, flexible
  – interaction entails change
Idiom: **Realign**

- stacked bars
  - easy to compare
    - first segment
    - total bar
- align to different segment
  - supports flexible comparison

**System: LineUp**

Idiom: **Animated transitions**

- smooth transition from one state to another
  - alternative to jump cuts
  - support for item tracking when amount of change is limited
- example: multilevel matrix views
- example: animated transitions in statistical data graphics
  - [https://vimeo.com/19278444](https://vimeo.com/19278444)

Manipulate

_change over time_

Select

Navigate

item reduction

_attribute reduction_

attribute reduction
Further reading

  —Chap 11: Manipulate View


Outline

• **Session 1: Principles** 9:15-10:30am
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling** 10:50-11:40am
  – Manipulate: Change, Select, Navigate
  – **Facet**: Juxtapose, Partition, Superimpose
  – Reduce: Filter, Aggregate

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
How to handle complexity: 3 more strategies

### Manipulate

- **Change**
  - Juxtapose
  - Superimpose

- **Select**

- **Navigate**

### Reduce

- **Filter**
- **Aggregate**
- **Embed**

### Derive

- Derive

- facet data across multiple views
Facet

- **Juxtapose**

- **Partition**

- **Superimpose**

- **Coordinate Multiple Side By Side Views**
  - Share Encoding: Same/Different
    - Linked Highlighting
  - Share Data: All/Subset/None
  - Share Navigation
Idiom: **Linked highlighting**

- see how regions contiguous in one view are distributed within another
  - powerful and pervasive interaction idiom

- encoding: different
  - *multiform*

- data: all shared

Idiom: **bird’s-eye maps**

- encoding: same
- data: subset shared
- navigation: shared
  - bidirectional linking

- differences
  - viewpoint
  - (size)

- overview-detail

*System: Google Maps*

Idiom: **Small multiples**

- **encoding**: same
- **data**: none shared
  - different attributes for node colors
  - (same network layout)
- **navigation**: shared

---

**System: Cerebral**

### Coordinate views: Design choice interaction

<table>
<thead>
<tr>
<th>Encoding</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>Subset</td>
</tr>
<tr>
<td>Same</td>
<td>Redundant</td>
</tr>
<tr>
<td>Different</td>
<td>Multiform</td>
</tr>
</tbody>
</table>

**why juxtapose views?**

- benefits: eyes vs memory
  - lower cognitive load to move eyes between 2 views than remembering previous state with single changing view
- costs: display area, 2 views side by side each have only half the area of one view
Idiom: Animation (change over time)

- weaknesses
  - widespread changes
  - disparate frames

- strengths
  - choreographed storytelling
  - localized differences between contiguous frames
  - animated transitions between states
Partition into views

- how to divide data between views
  - encodes association between items using spatial proximity
  - major implications for what patterns are visible
  - split according to attributes

- design choices
  - how many splits
    - all the way down: one mark per region?
    - stop earlier, for more complex structure within region?
  - order in which attribs used to split
Partitioning: List alignment

- single bar chart with grouped bars
  - split by state into regions
    - complex glyph within each region showing all ages
  - compare: easy within state, hard across ages

- small-multiple bar charts
  - split by age into regions
    - one chart per region
  - compare: easy within age, harder across states
Partitioning: Recursive subdivision

- split by type
- then by neighborhood
- then time
  - years as rows
  - months as columns

System: HIVE

Partitioning: Recursive subdivision

- switch order of splits
  - neighborhood then type
- very different patterns

Partitioning: Recursive subdivision

- different encoding for second-level regions
  - choropleth maps

Superimpose layers

• **layer**: set of objects spread out over region
  – each set is visually distinguishable group
  – extent: whole view

• design choices
  – how many layers?
  – how are layers distinguished?
  – small static set or dynamic from many possible?
  – how partitioned?
    • heavyweight with attrs vs lightweight with selection

• distinguishable layers
  – encode with different, nonoverlapping channels
    • two layers achievable, three with careful design
Static visual layering

• foreground layer: roads
  – hue, size distinguishing main from minor
  – high luminance contrast from background

• background layer: regions
  – desaturated colors for water, parks, land areas

• user can selectively focus attention
• “get it right in black and white”
  – check luminance contrast with greyscale view

Superimposing limits

- few layers, but many lines
  - up to a few dozen
  - but not hundreds

- superimpose vs juxtapose: empirical study
  - superimposed for local visual, multiple for global
  - same screen space for all multiples, single superimposed
    - tasks
      - local: maximum, global: slope, discrimination

Dynamic visual layering

• interactive, from selection
  – lightweight: click
  – very lightweight: hover

• ex: 1-hop neighbors

Further reading

  • Chap 12: Facet Into Multiple Views


Outline

• **Session 1: Principles  9:15-10:30am**
  – Analysis: What, Why, How
  – Marks and Channels, Perception
  – Color

• **Session 2: Techniques for Scaling  10:50-11:40am**
  – Manipulate: Change, Select, Navigate
  – Facet: Juxtapose, Partition, Superimpose
  – **Reduce: Filter, Aggregate**

http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney
How to handle complexity: 3 more strategies

- **Manipulate**
  - Change
  - Select
  - Navigate

- **Facet**
  - Juxtapose
  - Partition
  - Superimpose

- **Reduce**
  - Filter
  - Aggregate
  - Embed

- **Derive**

  - Derive

- Additional strategy:

  - reduce what is shown within single view
Reduce items and attributes

- reduce/increase: inverses
- filter
  - pro: straightforward and intuitive
    - to understand and compute
  - con: out of sight, out of mind
- aggregation
  - pro: inform about whole set
  - con: difficult to avoid losing signal
- not mutually exclusive
  - combine filter, aggregate
  - combine reduce, facet, change, derive
Idiom: **histogram**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - new table: keys are bins, values are counts
- bin size crucial
  - pattern can change dramatically depending on discretization
  - opportunity for interaction: control bin size on the fly
Idiom: **boxplot**

- static item aggregation
- task: find distribution
- data: table
- derived data
  - 5 quant attrs
    - median: central line
    - lower and upper quartile: boxes
    - lower upper fences: whiskers
      - values beyond which items are outliers
  - outliers beyond fence cutoffs explicitly shown

[40 years of boxplots. Wickham and Stryjewski. 2012. had.co.nz]
Idiom: Hierarchical parallel coordinates

- dynamic item aggregation
- derived data: hierarchical clustering
- encoding:
  - cluster band with variable transparency, line at mean, width by min/max values
  - color by proximity in hierarchy

Dimensionality reduction

• attribute aggregation
  – derive low-dimensional target space from high-dimensional measured space
  – use when you can’t directly measure what you care about
    • true dimensionality of dataset conjectured to be smaller than dimensionality of measurements
    • latent factors, hidden variables

Tumor Measurement Data

data: 9D measured space

\[ \text{DR} \]

derived data: 2D target space
Idiom: **Dimensionality reduction for documents**

**Task 1**
- **Input:** HD data
- **Output:** 2D data
- **What?** In high-dimensional data
- **Why?** Produce
- **What?** In 2D data
- **Why?** Derive

**Task 2**
- **Input:** 2D data
- **Output:** Scatterplot Clusters & points
- **What?** In 2D data
- **Why?** Discover
- **What?** In Scatterplot Clusters & points
- **Why?** Explore
- **How?** Encode
- **Identify**
- **Select**

**Task 3**
- **Input:** Scatterplot Clusters & points
- **Output:** Labels for clusters
- **What?** In Scatterplot Clusters & points
- **Why?** Produce
- **What?** In Labels for clusters
- **Why?** Annotate
Further reading

  – Chap 13: Reduce Items and Attributes


More Information

• this talk
  http://www.cs.ubc.ca/~tmm/talks.html#vad17sydney

• book page (including tutorial lecture slides)
  http://www.cs.ubc.ca/~tmm/vadbook
  – 20% promo code for book+ebook combo: HVN17

  – illustrations: Eamonn Maguire

• papers, videos, software, talks, full courses
  http://www.cs.ubc.ca/group/infovis
  http://www.cs.ubc.ca/~tmm

@tamaramunzner